U.S. DEPARTMENT OF COMMERCE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NATIONAL WEATHER SERVICE
NATIONAL METEOROLOGICAL CENTER

OFFICE NOTE 295

(Revised)

The Preparation of Binary Data for Transmission via The National Weather Service (Office of Systems Operations) Telecommunications Gateway (NWSTG)

John D. Stackpole

Automation Division

July 1988

NMC Office Note 295
(Revised)

John D. Stackpole
Automation Division

The Preparation of Binary Data for Transmission via The National Weather Service (Office of Systems Operations) Telecommunications Gateway (NWSTG)

I. Who is this note for?

This note is addressed to software engineers who may be called upon to prepare bulletins for national or international transmission of NMC generated data in binary, or "bit transparent" form. Binary transmission, as opposed to coded (or character) transmission, means that the communications devices will consider the entire bulletin as a stream of (binary) bits and transmit them as given. There may well be ASCII (or EBCDIC) characters imbedded within the transmitted data but they are not given any preferential treatment in the transmission process.

II. What must you construct?

A. The Queue Descriptor.

In addition to preparing the actual bulletin or bulletins to be transmitted, the software engineer must precede each bulletin with an 80-byte "queue descriptor." This queue descriptor contains information needed by the communications people and includes empty space for the communications people to use internally. When the bulletin is eventually transmitted the queue descriptor will have been stripped off and replaced by a 16-byte "Communications Header", described below. The communications header will be received by the user who therefore should be told to expect it.

Here are the required contents of the queue descriptor:

Queue Descriptor

Bytes	Content
1-16	"QUEUE DESCRIPTOR" in EBCDIC characters (EBCDIC is the standard IBM/NAS machine representation of characters).
17-21	Fill with integer zeroes (the comms people will use this space).

22	Number of 80-byte increments in bulletin (binary integer).
23	Number of bytes in last increment (binary integer). See below for explanation of bytes 22 and 23.
24-28	Fill with integer zeroes (more comms space).
29-34	The 6-character bulletin name in EBCDIC. This same name is presumably also to be placed right at the start of the actual bulletin (but in ASCII characters there). Subroutines W3AI38 and W3AI39 will convert EBCDIC to ASCII, and vice versa.
35-38	Day of month and UTC (Z) hours; two EBCDIC characters each. This information, probably describing the observation time of data, is most likely also to be placed at the beginning of the actual bulletin (in ASCII).
39-40	The hour and minute time of creation of the bulletin. (The system clock will give you this time - use W3FQ02). The hours and minutes are represented in the two bytes as four-bit BCD - subroutine W3AQ15 will manufacture and place them in bytes 39 and 40 for you.
41-45	Transmission Catalog Number (CLLLL) in EBCDIC. The CLLLL has fallen into disuse in the communications world. For now, set the CLLLL = "55555".
46-80	Fill with integer zeroes.

Explanation of Bytes 22 and 23

To arrive at the numbers that are to be placed in those locations, first figure the total length (in bytes) of the bulletin excluding the 80-byte queue descriptor, excluding the Communications Header, but including the 21 byte WMO Header (see below). Divide this total number of bytes by 80, round up, if necessary, to the next integer to determine how many 80-byte "increments" are needed to contain the entire bulletin. This number goes into byte 22. The last increment may not be full - the remainder after the division will tell you how many bytes of bulletin data (from 1 to 79) are in that last increment; that number goes into byte 23. If the remainder is zero, then the last increment is exactly full and 80 should be placed into byte 23.

B. The Communications Header

As mentioned above, the preparer of the bulletin does not construct this 16-byte header; the communications people put it there after removing the 80-byte queue descriptor. However the header does go out as the first part of the message so you had better know about it and tell your recipient to expect it.

Communications Header

Byte	Content
1	"SOH" - the ASCII start of header symbol: (01)hex.
2-4	CRCRLF - two ASCII carriage returns and one line feed: (ODODOA)hex.
5-7	Communications Sequence Number.
8	An ASCII blank character: (20)hex
9-13	Transmission Catalogue Number (CLLLL).
14-16	CRCRLF (as above).

The communications sequence and transmission catalogue numbers are mainly for local use and may be ignored. Indeed the entire communications header can be ignored by the end user just as long as he knows those bytes are there.

C. The WMO Header

The first portion of the "real" bulletin has a more or less standard form for the WMO global telecommunications system. This is the portion of the material directly following the 80 byte queue descriptor that you must put together.

WMO Header

Bytes	Content
1-6	Bulletin name. Six ASCII characters. Bulletin names are supplied, upon request, by the Office of Systems Operation, Telecommunications and Dissemination Branch. The names often reflect the content of the messages.
7	An ASCII blank character: (20)hex.

8-11	"KWBC": 4 ASCII characters. This is the standard identifier for NWS generated bulletins.
12	Blank: (20)hex.
13-14	Day of month (2 ASCII numeric characters).
15-18	Time: hour, minute (4 ASCII numeric characters).
19-21	CRCRLF (As above in the communications header).

The day and time usually apply to the observation time of the data to follow, the initial time of the forecast to follow, or whatever.

From byte 22 on, you are on your own.

D. The Bulletin itself.

The content of the bulletin is, of course, quite variable and is, or should be, designed to maximize information transfer. Such formats as "BUFR" and "GRIB" have been designed to compress out redundancy as much as possible without sacrificing precision. For reasons related to some message switching centers' current limitations, a single bulletin must not exceed 13,500 bytes in length.

If you have extensive data that require the use of many bulletins, no problems. You do have to have unique Bulletin Names for each bulletin and <u>each</u> bulletin must have its own 80-byte queue descriptor prefixed to it prior to handing off the set of bulletins (one or more) to the NWSTG machines.

III. <u>How to pass the bulletins to the NWSTG communicators</u>

Unfortunately even though you have constructed your bulletins, each complete with its queue descriptor in core, you still have to accommodate the communications people in the way you write your bulletins to a disk file for them to pick up.

1) Your bulletins must be written as one or more physical records (blocks) of length 1280 bytes. If your bulletin is less than 1280 bytes (or any continuation block is only partially used) the remainder must be filled with EBCDIC blanks, (40)hex. Subroutine W3AI19 can be used to help you with this blocking-with-blank-fill business. Once a block (or set of NN blocks) has been created, the following code fragment will do the actual writing of one or more blocks for you:

```
      DIMENSION NBLCK(320,NNN)
      .
      .
      .
      WRITE (fmt,unit) (NBLCK(1,N), N=1,NN)
fmt   FORMAT(2(160A4))

//FTunitF001   DD DSN=yourdataset,DISP=(NEW,KEEP),
                 UNIT=3375,VOL=SER=SWING2,
                 SPACE=(CYL,3),
                 DCB=(RECFM=F,BLKSIZE=1280)
```

NBLCK should be dimensioned large enough to hold at least NN times 320 4-byte words. In the DD card, the DSN should be selected in consultation with the Automation Division Production Management Branch as certain names have operational consequences; the "SPACE=" parameter should be set to a generous over-estimate of the necessary space (the file will be deleted automatically after your bulletins are sent on their way); the DCB parameter takes care of the actual blocking on the output device. Each new bulletin must start in a new physical record. The trailing EBCDIC blanks will be stripped off when the bulletin is transmitted.

2) After the last block of the last bulletin of a series you must write one more 1280-byte block, a terminator. It has to look like this:

Terminator

Bytes	Content
1-8	"XTRN END", in EBCDIC
9-1280	Anything

3) After all the blocks are written to the output data set on the SWING2 disk, the data set must be closed. You can use a FORTRAN REWIND statement or, in VS FORTRAN, a CLOSE statement.

4) After the data set is closed you have to notify the NWSTG system that the message(s) are ready on SWING2. What you do is call subroutine W3AG15 with "XTRN" in the second argument of the call sequence. See the DOCBLOCK for W3AG15 for more details on that process.

And that's it - your bulletins are on their way.

IV. Checkout Considerations

All of the above assumes that your code is working properly; unless you are some kind of superbeing, it won't be for the first few times. During your initial checkout efforts please do not write anything on VOL=SER=SWING2, but use a disk pack of your own, or a temporary data set (DSN=&&TEMP), with DISP=(NEW,PASS). In

checkout runs, W3AG15 will not send spurious notifications to the NWSTG machines.

To aid in your checkout work, R. Allard has written a very handy utility code that will dump, in hexadecimal, the content of the queue descriptor and your bulletin. Invoke his code like this:

```
//DOIT     EXEC PGM=XTRNPRNT
//STEPLIB    DD DSN=NMC.PROD.MVSUTL.LOAD,DISP=SHR
//FT06F001   DD SYSOUT=*
//FT88F001   DD DSN=&&TEMP,DISP=(OLD,DELETE)
```

where &&TEMP contains the bulletin(s) written in a previous step.

Once you are satisfied that your bulletins are properly constructed, you can make a final end-to-end test by changing the first 11 characters of the WMO header to "CHEK34 OUTT", sending the output to SWING2 as indicated above, and using a "production-style" job name for your test. A production job name is one with a numeric character in positions 3 and 8 of the JOBNAME (e.g., WW2JDSX1). The special header will cause your bulletin to travel all the way through the NWSTG machines, leaving a normal audit trail, but, in the end, go nowhere. Contact the communications people (Dan Starosta is a good one to start with) within a couple of hours after your test run and have him check that your bulletin did indeed arrive and go where it was supposed to go.

Don't forget to restore the WMO header to its proper form before submitting your job for operational implementation.

U.S. DEPARTMENT OF COMMERCE
NATIONAL OCEANIC AND ATMOSPHERIC ADMINISTRATION
NATIONAL WEATHER SERVICE
NATIONAL METEOROLOGICAL CENTER

OFFICE NOTE 295

(Revised)

The Preparation of Binary Data for Transmission via The National Weather Service (Office of Systems Operations) Telecommunications Gateway (NWSTG)

Mary Ann Farley
&
John D. Stackpole

Automation Division

March 1994

NMC Office Note 295
(Revised)

Mary Ann Farley & John D. Stackpole
Automation Division

The Preparation of Binary Data for Transmission via The National Weather Service (Office of Systems Operations) Telecommunications Gateway (NWSTG)

I. Who is this note for?

This note is addressed to software engineers who may be called upon to prepare bulletins for national or international transmission of NMC generated data in binary, or "bit transparent" form. Binary transmission, as opposed to coded (or character) transmission, means that the communications devices will consider the entire bulletin as a stream of (binary) bits and transmit them as given. There may well be ASCII (or EBCDIC) characters imbedded within the transmitted data but they are not given any preferential treatment in the transmission process.

II. What must you construct?

A. The Queue Descriptor.

In addition to preparing the actual bulletin or bulletins to be transmitted, the software engineer must precede each bulletin with an 80-byte "queue descriptor." This queue descriptor contains information needed by the communications people and includes empty space for the communications people to use internally. When the bulletin is eventually transmitted the queue descriptor will have been stripped off and replaced by a 10-byte "Communications Header", described below. The communications header will be received by the user who therefore should be told to expect it.

A utility subroutine, W3FI62, is available in the NMC W3 library which will generate the 80-byte "queue descriptor." The following explanation is provided to give a thorough explanation of the "queue descriptor."

Queue Descriptor

Bytes	Content
1-16	"QUEUE DESCRIPTOR" in EBCDIC characters (EBCDIC is the standard IBM/NAS/HDS machine representation of characters).
17-20	Fill with integer zeroes (the comms people will use this space).
21-22	Number of 80-byte increments in bulletin (binary integer).
23	Number of bytes in last increment (binary integer). See below for explanation of bytes 21-22 and 23.
24-28	Fill with integer zeroes (more comms space).

29-34	The 6-character bulletin name in EBCDIC. This same name is presumably also to be placed right at the start of the actual bulletin (but in ASCII characters there). Subroutines W3AI38 and W3AI39 will convert EBCDIC to ASCII, and vice versa.
35-38	Day of month and UTC (Z) hours; two EBCDIC characters each This information, probably describing the observation time of data, is most likely also to be placed at the beginning of the actual bulletin (in ASCII).
39-40	The hour and minute time of creation of the bulletin. (The system clock will give you this time - use W3FQ02). The hours and minutes are represented in the two bytes as four-bit BCD -- subroutine W3AQ15 will manufacture and place them in bytes 39 and 40 for you.
41-45	Transmission Catalog Number (CLLLL) in EBCDIC The CLLLL has fallen into disuse in the communications world. For now, set the CLLLL = "55555".
46-80	Fill with integer zeroes.

Explanation of Bytes 21-22 and 23

To arrive at the numbers that are to be placed in those locations, first figure the total length (in bytes) of the bulletin excluding the 80-byte queue descriptor, excluding the Communications Header, but including the 21 byte WMO Header (see below). Divide this total number of bytes by 80, round up, if necessary, to the next integer to determine how many 80-byte "increments" are needed to contain the entire bulletin. This number goes into bytes 21-22. The last increment may not be full - the remainder after the division will tell you how many bytes of bulletin data (from 1 to 79) are in that last increment; that number goes into byte 23. If the remainder is zero, then the last increment is exactly full and 80 should be placed into byte 23.

B. The Communications Header

As mentioned above, the preparer of the bulletin does not construct this 10-byte header; the communications people put it there after removing the 80-byte queue descriptor. However the header does go out as the first part of the message so you had better know about it and tell your recipient to expect it.

Communications Header

Byte	Content
1	"SOH" - the ASCII start of header symbol: (01)hex.

2-4	CRCRLF - two ASCII carriage returns and one line feed: (ODODOA)hex.
5-7	Communications Sequence Number.
8-10	CRCRLF (as above).

The communications sequence number is mainly for local use and may be ignored. Indeed the entire communications header can be ignored by the end user just as long as he knows those bytes are there.

C. The WMO Header

The first portion of the "real" bulletin has a more or less standard form for the WMO global telecommunications system. This is the portion of the material directly following the 80 byte queue descriptor that you must put together.

WMO Header

Bytes	Content
1-6	Bulletin name. Six ASCII characters. Bulletin names are supplied, upon request, by the Office of Systems Operation, Telecommunications and Dissemination Branch. The names often reflect the content of the messages.
7	An ASCII blank character: (20)hex.
8-11	"KWBC": 4 ASCII characters. This is the standard identifier for NWS generated bulletins.
12	Blank: (20)hex.
13-14	Day of month (2 ASCII numeric characters).
15-18	Time: hour, minute (4 ASCII numeric characters).
19-21	CRCRLF (As above in the communications header).

The day and time usually apply to the observation time of the data to follow, the initial time of the forecast to follow, or whatever.

From byte 22 on, you are on your own.

D. The Bulletin itself.

The content of the bulletin is, of course, quite variable and is, or should be, designed to maximize information transfer. Such formats as "BUFR" and "GRIB" have been designed to compress out redundancy as much as possible without sacrificing precision. For reasons related to some message switching centers' limitations, a single bulletin must not exceed 13,500 bytes in length. However, the communications folks will accept large bulletins and split them up for you,

using some extensions to the WMO header to identify the parts. It will be up to the communications people at the other end to put humpty-dumpty back togeter again.

If you have extensive data that require the use of many bulletins, no problems. You do have to have unique Bulletin Names for each bulletin and each bulletin must have its own 80-byte queue descriptor prefixed to it prior to handing off the set of bulletins (one or more) to the NWSTG machines.

III. How to pass the bulletins to the NWSTG communicators

Unfortunately even though you have constructed your bulletins, each complete with its queue descriptor in memory, you still have to accommodate the communications people in the way you write your bulletins to a disk file for them to pick up.

1) Your bulletins must be written as one or more physical records (blocks) of length 1280 bytes. If your bulletin is less than 1280 bytes (or any continuation block is only partially used) the remainder must be filled with EBCDIC blanks, (40)hex. Subroutine W3AI19 can be used to help you with this blocking-with-blank-fill business. Once a block (or set of NN blocks) has been created, the following code fragment will do the actual writing of one or more blocks for you:

```
                DIMENSION NBLCK(320,NNN)
                .
                .
                .
                WRITE (fmt,unit) (NBLCK(1,N), N=1,NN)
          fmt   FORMAT(2(160A4))

//FTunitF001  DD DSN=yourdataset,DISP=(NEW,KEEP),
                  UNIT=3380,VOL=SER=NMCSW2,
                  SPACE=(CYL,3),
                  DCB=(RECFM=FB,LRECL=1280,BLKSIZE=12800)
```

NBLCK should be dimensioned large enough to hold at least NN times 320 4-byte words. In the DD card, the DSN should be selected in consultation with the Automation Division Production Management Branch as certain names have operational consequences; the "SPACE=" parameter should be set to a generous over-estimate of the necessary space (the file will be deleted automatically after your bulletins are sent on their way); the DCB parameter takes care of the actual blocking on the output device. Each new bulletin must start in a new physical record. The trailing EBCDIC blanks will be stripped off when the bulletin is transmitted.

2) After the last block of the last bulletin of a series you must write one more 1280-byte block, a terminator. It has to look like this:

Terminator

Bytes	Content
1-8	"XTRN END", in EBCDIC
9-1280	Anything

3) After all the blocks are written to the output data set on the NMCSW2 disk, the data set must be closed. You can use a FORTRAN REWIND statement or, in VS FORTRAN, a

CLOSE statement.

4) After the data set is closed you have to notify the NWSTG system that the message(s) are ready on NMCSW2. What you do is call subroutine W3AG15 with "XTRN" in the second argument of the call sequence. See the DOCBLOCK for W3AG15 for more details on that process.

And that's it - your bulletins are on their way.

IV. Checkout Considerations

All of the above assumes that your code is working properly; unless you are some kind of superbeing, it won't be for the first few times. During your initial checkout efforts please do not write anything on VOL=SER=NMCSW2, but use a disk pack of your own, or a temporary data set (DSN=&&TEMP), with DISP=(NEW,PASS). In checkout runs, W3AG15 will not send spurious notifications to the NWSTG machines.

To aid in your checkout work, a very handy utility code is available that will dump, in hexadecimal, the content of the queue descriptor and your bulletin. Invoke the code like this:

```
//DOIT      EXEC PGM=XTRNPRNT
//STEPLIB   DD   DSN=NMC.PROD.MVSUTL.LOAD,DISP=SHR
//FT06F001  DD   SYSOUT=*
//FT88F001  DD   DSN=&&TEMP,DISP=(OLD,DELETE)
```

where &&TEMP contains the bulletin(s) written in a previous step.

Once you are satisfied that your bulletins are properly constructed, you can make a final end-to-end test by changing the first 11 characters of the WMO header to "CHEK34 OUTT", sending the output to NMCSW2 as indicated above, and using a "production-style" job name for your test. A production job name is one with a numeric character in positions 3 and 8 of the JOBNAME (e.g., WW2MAFX1). The special header will cause your bulletin to travel all the way through the NWSTG machines, leaving a normal audit trail, but, in the end, go nowhere. Contact the communications people within a couple of hours after your test run and have them check that your bulletin did indeed arrive and go where it was supposed to go.

Don't forget to restore the WMO header to its proper form before submitting your job for operational implementation.